

# Presenting a CRASH COURSE IN AGILE TECHNICAL PRACTICES

## CONTINUOUS INTEGRATION (CI)

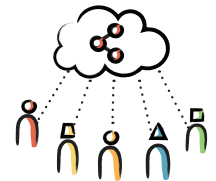
The act of merging code into the source repository frequently (measured in minutes and hours).



*Continuously merging resolves conflicts early, allowing many developers to work simultaneously.*

- ✓ Increased Productivity
- ✓ Reduced Rework

## IN OTHER WORDS...



*File share vs. google docs*

## TEST DRIVEN DEVELOPMENT (TDD)

Tests are written first, then just enough code is written to pass. It is then refactored to ensure maintainability.



*Defining the answer before the question reduces code bloat and provides easier maintenance.*

- ✓ Increased Speed
- ✓ Increased Quality
- ✓ Maintainability



*Mould*

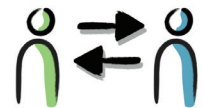
## PAIR PROGRAMMING

A technique in which two programmers sit together to design and produce code. As one writes, the other guides and reviews each line. Roles switch frequently.



*Pairing puts two brains on the problem from start to finish—resulting in faster development and better solutions.*

- ✓ Increased Speed
- ✓ Increased Quality
- ✓ Clean Code
- ✓ Better Solutions



*Co-pilots*

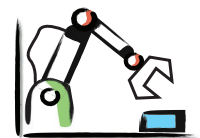
## AUTOMATED TESTING

Wherever possible, allow the computer to do the work and remove human error.



*Ensure that all previously written code still works through comprehensive automated execution of regression testing.*

- ✓ Increased Speed
- ✓ Increased Quality/Stability
- ✓ Reduce Manual Intervention



*Automation and robotics*

Become a  
**CERTIFIED SCRUM DEVELOPER®**  
IN 3-5 DAYS!

Learn more at [projectbrilliant.com/trainings](https://projectbrilliant.com/trainings)